REMARKS/ARGUMENTS

Claims 1-69, 135 and 136 are pending in the present application. Claims 1, 34, and 135 are amended. Claims 70-134 and 137-140 were previously canceled. Support for the amendments can be found in the specification on page 13, lines 30-38; page 14, lines 8-12; and page 14, lines 32-39. Reconsideration of the claims is respectfully requested.

I. Claim Objections

I.A. First Objection

The Examiner objects to claims 1, 34, and 35 on three bases. Applicants traverse these objections.

First, the examiner objects to these claims on the basis that the term "adapted to transform" is not disclosed sufficiently in the specification. Specifically, the examiner states that:

> Claims 1, 34, and 135 are objected to because of the following informalities: the use of the term 'adapted to transform' (cl. 1, lines 10, 12; cl. 34, line 13; cl. 135, li. 10-11) should be corrected to map the teachings from the specifications. According to which, the derived graphical information (Fig. 9-11) resulting from mapping of data to a database are used by the interface and the XSL support tool to generate target XML files. The level accomplishment by any entity being 'adapted to transform' is not disclosed sufficiently in the Specifications because for this to be visible a clear definition of any adaptation ('adapted to transform') should be explicit. There is difference between a feature being adapted to and being actually used or performing; thus, the claim does not lay out a clear meaning as to whether a transformation is taken place when 'adapted to' is not teaching an achieved execution by an action.

Office Action of September 1, 2006, p. 2.

Claim 1 as amended is as follows:

> 1. A computer implemented method for deriving transformations for transforming data from one data schema to another, comprising:
> receiving a source data schema and a target data schema, the source data schema being different from the target data schema;
> mapping the source data schema into an ontology model;
> mapping the target data schema into the ontology model; and
> deriving, using the ontology model, a derived transformation for transforming first data conforming to the source data schema into second data conforming to the target data schema, wherein the derived transformation can transform the first data directly into the second data, and wherein the derived transformation can transform the first data into the second data without transforming the first data via an ontological format.

In response, Applicants have amended claim 1 to strike the words "is adapted to" and replace that term with the word "can." Support for this amendment can be found on page 13, lines 30-38; page 14,

lines 8-12; and page 14, lines 32-39. The specification clearly teaches that the transformation can transform a first data into a second data and that the transformation can transform the first data into the second data without transforming data via an ontological format. Thus, this objection is overcome.

Claims 34 and 135 contain similar features. Therefore, the objection to claims 1, 34, and 135 has been overcome.

I.B. Second Objection

Next, the examiner objects to claims 1, 34, and 135 on the basis that the term "derived transformation" is hard to understand and obscure. Specifically, the examiner states that:

> Further, the above claim language needs to avoid reciting hard-to-understand limitation. That is, the claim as of now amount to reciting that a *derived transformation* would be purported for transforming a first data into a second data, which seems obscure as to why a transformation can in turn become the active entity transforming something else. The above terminology should be corrected lest its interpretation would lead onto the deficiency of the non-statutory type subject matter.

Office Action of September 1, 2006, p. 2 (emphasis in original).

Claim 1 as amended is as follows:

> 1. A computer implemented method for deriving transformations for transforming data from one data schema to another, comprising:
>
> receiving a source data schema and a target data schema, the source data schema being different from the target data schema;
>
> mapping the source data schema into an ontology model;
>
> mapping the target data schema into the ontology model; and
>
> deriving, using the ontology model, a derived transformation for transforming first data conforming to the source data schema into second data conforming to the target data schema, wherein the derived transformation can transform the first data directly into the second data, and wherein the derived transformation can transform the first data into the second data without transforming the first data via an ontological format.

The objection is without foundation. The term "derived transformation" is clearly delineated in the specification and clearly understandable to one of ordinary skill in the art. For example, see page 14, line 9 where the term "derived transformations" is explicitly used in claim 1. Other examples of using a "derived transformation" exist throughout the specification.

Additionally, the examiner's assertion that the term would be "obscure as to why a transformation can in turn become the active entity transforming something else," is incorrect and without foundation. A "transformation" is a known concept in which a mathematical function can be used to transform one object into another object.

In one non-limiting illustrative example of the claimed invention, first and second data schema are mapped into an ontology model. Using the ontology model, a "derived transformation" is derived. The "derived transformation" can transform first data directly into second data. The "derived transformation" can transform the first data into the second data without transforming the first data via an ontological format. Thus, in this example, an ontology model is used to derive a transformation, but the transformation itself does not use an ontological format. This claimed process is clearly understandable to one of ordinary skill in the art.

Because the process in claim 1 is clearly understandable and because the specification unambiguously uses the claimed terms, no basis exists to object to the claimed term "derived transformations." Accordingly, this objection is overcome.

Claims 34 and 135 contain similar features. Therefore, the objection to claims 1, 34, and 135 has been overcome.

I.C. Third Objection

Next, the examiner objects to claims 1, 34, and 135 on the basis that the term "via an ontological format" has no clear definition. Specifically, the examiner states that:

> Claims 1, 34, and 135 are objected to because of the language reciting: 'deriving using the ontology model...a transformation for transforming' in conjunction with 'to transform...without transforming the first data via an ontological format'. There is no clear definition of the phrase 'via an ontological format' in light of the ontology-oriented transformation/mapping process as disclosed in the Specifications (e.g. pg. 14- 16). That is, according to which, the RDBS structures are extracted into basic elements displayed as chart or table reflecting the database row/columns elements and the transformation from source data into target data is based on such mapping as extracted to enable creation of a form of model in terms of *00* class or objects from which to apply further enhancement by the users (*); whereby it is not reasonably conveyed that a format such as a ontological representation is not present during the transformation process. From the claim, the transformation of the first data is construed as being implemented not 'via' an ontology format. When there is no clear explicit definition in the disclosure as to what this 'without transforming... via an ontological format', this *ontological format* negative limitation would be unclear as to whether it entails a format of transformed data intermediate to the final result or merely any graphical format used during the realization of the derived data, an example of which being the representation of *00* classes and attributes based on mapping results, in which case the *00* class are considered a graphical format pertaining to a model or some high-level of abstraction, i.e. an ontology format. But reciting 'using the ontology model' for transformation along with this transformation being not done using such ontology format would rather be hard to construe in light of the analysis in (*) and of the *00* class high-level representation (otherwise perceived as ontological view). Absent a definition of a transformation using --or not using -- an ontology format, the disclosure lacks sufficient, deliberate and clear teaching as to how this non-use of ontological format has been

> implemented. The language of the 'without transforming via... format' should be readjusted to also obviate a 112 type of rejection, mostly because ontology signifies a study using high-level representation of interacting elements, and the Specifications does teach a combination of 00 classes format that represent a model (see Fig. 1) which reasonably contradicts with the above claimed ontological format non-use limitation.

Office Action dated September 1, 2006, pp. 2-4 (emphasis in original).

Claim 1 as amended is as follows:

> 1. A computer implemented method for deriving transformations for transforming data from one data schema to another, comprising:
>
> receiving a source data schema and a target data schema, the source data schema being different from the target data schema;
>
> mapping the source data schema into an ontology model;
>
> mapping the target data schema into the ontology model; and
>
> deriving, using the ontology model, a derived transformation for transforming first data conforming to the source data schema into second data conforming to the target data schema, wherein the derived transformation can transform the first data directly into the second data, and wherein the derived transformation can transform the first data into the second data without transforming the first data via an ontological format.

The objection is without foundation. The phrase "via an ontological format," in conjunction with the phrase "without transforming," is clearly delineated in the specification and clearly understandable to one of ordinary skill in the art. For example, see page 14, lines 10-11 in which these phrases are explicitly used. Other examples of use of these terms appear throughout the specification.

Additionally, the examiner's assertion that "this ontological format negative limitation would be unclear as to whether it entails a format of transformed data intermediate to the final result or merely any graphical format used during the realization of the derived data," is incorrect and without foundation. The term "ontological format" is a known concept and is described in excruciating detail in the specification. See, for example, pages 1-5, particularly pages 3-5, in which an ontological model is described in exact mathematical terms. From the ontological model one of ordinary skill would understand an ontological format. In addition, the term "ontological format" is explicitly used in the specification, see page 14, line 11 and other locations, in relation to the other claim language.

In one non-limiting illustrative example of the claimed invention, first and second data schema are mapped into an ontology model. Using the ontology model, a "derived transformation" is derived. The "derived transformation" can transform first data directly into second data. The "derived transformation" can transform the first data into the second data without transforming the first data via an ontological format. Thus, in this example, an ontology model is used to derive a transformation, but the transformation itself does not use an ontological format. This claimed process is clearly understandable to one of ordinary skill in the art.

Because the claimed process is clearly understandable and because the specification unambiguously uses the claimed terms, no basis exists to object to the claimed term "via an ontological format." Accordingly, this objection is overcome.

Claims 34 and 135 contain similar features. Therefore, the objection to claims 1, 34, and 135 has been overcome.

II. Asserted Indefiniteness, 35 U.S.C. §112, Second Paragraph

The Examiner rejected claims 1, 34, and 135 as indefinite under 35. U.S.C. §112, Second Paragraph. This rejection is respectfully traversed.

Applicants first address the rejection of claim 1. The Examiner states that:

> The reciting of 'without transforming via an ontological format' is not conveyed clearly in the disclosure or the claim to convey to one skill in the art how transformation is done such that it necessarily obviate the stage using a ontological format. From the Specifications, ontological structure is one model where relationship among elements are visible; and in view of the analysis from above in the claim Objections, the derived class objects depicted in the user graphical for the user to further modify the model does read on a format that is ontological, and based on the Specifications from Fig. 1, this model enable transformation in the last step. It is therefore unclear as to how transformation of the source data into some form can be achieved when it is required that there is no ontological format involved in this transformation. One skill in the art would not be able to construe the claimed invention based on what appears to be incompatible teachings from claim interpretation and the very facts from the disclosure; and this has been mentioned in the above Claim Objection, so that as a whole the claims (1, 34, 135) would make it hard for one skill in the art to make use of the invention.

Office Action dated September 1, 2006, pp. 4-5.

Claim 1 as amended is as follows:

> 1. A computer implemented method for deriving transformations for transforming data from one data schema to another, comprising:
> receiving a source data schema and a target data schema, the source data schema being different from the target data schema;
> mapping the source data schema into an ontology model;
> mapping the target data schema into the ontology model; and
> deriving, using the ontology model, a derived transformation for transforming first data conforming to the source data schema into second data conforming to the target data schema, wherein the derived transformation can transform the first data directly into the second data, and wherein the derived transformation can transform the first data into the second data without transforming the first data via an ontological format.

The rejection is without foundation. The phrase "without transforming via an ontological format," is clearly delineated in the specification and clearly understandable to one of ordinary skill in the

art. For example, see page 14, lines 10-11 in which this phrase is explicitly used. Other examples of use of this phrase appear throughout the specification.

Additionally, the examiner's assertion that "It is therefore unclear as to how transformation of the source data into some form can be achieved when it is required that there is no ontological format involved in this transformation," is incorrect and without foundation. In one non-limiting illustrative example of the claimed invention, first and second data schema are mapped into an ontology model. Using the ontology model, a "derived transformation" is derived. The "derived transformation" can transform first data directly into second data. The "derived transformation" can transform the first data into the second data without transforming the first data via an ontological format. Thus, in this example, an ontology model is used to derive a transformation, but the transformation itself does not use an ontological format when transforming data. This claimed process is clearly understandable to one of ordinary skill in the art and this claimed process is explicitly defined and well-described in the specification.

Because the claimed process is clearly understandable and because the specification unambiguously uses the claimed terms, no basis exists to object to the claimed phrase "without transforming via an ontological format." Accordingly, this rejection is overcome.

Claims 34 and 135 contain similar features. Therefore, the rejection of claims 1, 34, and 135 has been overcome.

III. <u>35 U.S.C. §102, Anticipation</u>

The Examiner rejected claims 1-8, 20, 24, 26-41, and 135-136 as anticipated by *Wachtel*, <u>Method and Apparatus for Intelligent Data Assimilation</u>, U.S. Patent 6,847,974 (January 25, 2005) (hereinafter "*Wachtel*"). This rejection is respectfully traversed.

A prior art reference anticipates the claimed invention under 35 U.S.C. §102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). In this case each and every feature of the presently claimed invention is not identically shown in the cited reference, arranged as they are in the claims.

Applicants first address the rejection of claim 1. Claim 1 as amended is as follows:

> 1. A computer implemented method for deriving transformations for transforming data from one data schema to another, comprising:
>
> receiving a source data schema and a target data schema, the source data schema being different from the target data schema;
>
> mapping the source data schema into an ontology model;
>
> mapping the target data schema into the ontology model; and
>
> deriving, using the ontology model, a derived transformation for transforming first data conforming to the source data schema into second data conforming to the target data schema, wherein the derived transformation can transform the first data directly into the second data, and wherein the derived transformation can transform the first data into the second data without transforming the first data via an ontological format.

With regard to the rejection of claim 1 the examiner states that:

> **As per claim 1**, Wachtel discloses a method for deriving transformations for transforming data from one data schema to another, comprising:
>
> receiving a source data schema (e.g. Fig. 1 4) and a target data schema (e.g. Fig. 1 7 - Note: receiving a XML message in order to process it again reads on receiving a target schema - see col. 15, lines 1 -29), the source data being different from the target data (Note: samples of schemas in Fig. 14, 17 depicts distinct forms of XML);
>
> mapping the source data schema into an ontology model; mapping the target data schema into the ontology model (e.g. Fig. 6-9; *xml,* workflow, *map* - col. 5, line 62 to col. 7, line 14 - Note: XML information/metadata when parsed by XPATH into a tree reads on a schema of definition); and
>
> deriving a transformation (e.g. Fig. 8-9, col. 12, line 51 to col. 13, line 26 -Note: *Intelligent data assimilation system* reads on automatic derivation) for transforming data conforming to the source data schema into data conforming to the target data schema (e.g. Fig. 4- 5; Fig. 6-9; Fig. 17-1 8 - Note: derivation of class in a workflow - see Fig. 4-5 --view reads on derived transformation used for further transformation; and output response based on mapping against ontology metastore based on service and output requests - Fig. 14, 17 -- reads on source data schema in request conforming to target schema), using the ontology model;
>
> wherein the transformation transforms data conforming to source schema directly to data conforming target schema (Note: using abstracted concepts stored in the model and invoking one or more pre-stored templates to instantiate a workflow - see col. 5, line 46 to col. 7, line 14 - - to include intelligence enabling the transforming into a target schema reads on data schema from the request --i.e. source schema -- directly converted into output schema without any intermediate data schema format; Fig. 10, step 5 16, 5 18; Fig 6 and related text),
>
> wherein the transformation is adapted to transform the first data into the second data without transforming the first data via ontological format (Note: providing mapping by Weblogic assimilation integrator - see col. 9 and Fig. 4-- to obtain atomic objects and derived objects grouping into a model (or ontological

> abstraction) or LSO-based workflow, enhancing this LSO collection with object oriented attributes - see Fig. 9 - so to generate XML form to fulfill a request - see col. 7, lines 4-14; col. 14, bottom -- **reads on** without transforming via any ontology format, i.e. using no intermediate format from the DBMS, from the atomic objects, no intermediate format from the extracted basic/atomic units generated by the provider Weblogic mapping process - see col. 9).

Office Action dated September 1, 2006, pp. 5-7 (emphasis in original).

Wachtel does not anticipate claim 1 because *Wachtel* does not teach the feature of, "wherein the derived transformation can transform the first data into the second data without transforming the first data via an ontological format," as recited in claim 1. The examiner asserts otherwise, stating specifically that:

> (Note: providing mapping by Weblogic assimilation integrator - see col. 9 and Fig. 4-- to obtain atomic objects and derived objects grouping into a model (or ontological abstraction) or LSO-based workflow, enhancing this LSO collection with object oriented attributes - see Fig. 9 - so to generate XML form to fulfill a request - see col. 7, lines 4-14; col. 14, bottom -- **reads on** without transforming via any ontology format, i.e. using no intermediate format from the DBMS, from the atomic objects, no intermediate format from the extracted basic/atomic units generated by the provider Weblogic mapping process - see col. 9).

Office Action dated September 1, 2006, pp. 6-7 (emphasis in original).

The examiner's assertion that *providing mapping by Weblogic assimilation integrator to obtain atomic objects into an ontological abstraction* is equivalent to the claimed feature, "*wherein the derived transformation can transform the first data into the second data without transforming the first data via an ontological format*," is without foundation. The examiner's assertion is based on the examiner's flawed understanding that claim 1 somehow requires an ontological format when transforming first data to second data, given that the transformation is derived from an ontology model. Instead, in a non-limiting illustrative example explicitly provided in the specification, an ontology model is used to derive the transformation. The transformation itself does not use an ontological format.

In contrast, *Wachtel* teaches the following in the sections cited by the examiner:

> In one embodiment of an intelligent data assimilation system according to the present invention, the underlying platform for these components is a J2EE compatible application server. A suitable application server is BEA Weblogic Application Server. The capabilities of BEA's Weblogic Application Server include a distributed transaction manager, Java Messaging Services (JMS), support for extensible Markup Language (XML), and J2EE standards support.
>
> In one embodiment of an intelligent data assimilation system according to the present invention, an intelligent data assimilation system uses a BEA Weblogic Process Integrator (WLPI) to implement workflow tasks. WLPI provides the intelligent data assimilation system the ability to design and automate business processes that integrate data provider applications, search services, and human intervention.

In an embodiment of an intelligent data assimilation system according to the present invention, an Oracle DataBase Management System (DBMS) provides the intelligent data assimilation system with persistence capabilities.

FIG. 4 is a diagram of levels within an exemplary portion of one embodiment of an intelligent data assimilation system ontology according to the present invention. ***The intelligent data assimilation system ontology manages three levels of data abstraction.*** Initially, raw data fields 140 in a data set are positionally mapped from a data provider into atomic objects such as person atomic object 146 and credit atomic object 148. Atomic objects add type and basic naming information to the data fields. Intelligence about the meaning of the data fields included in the atomic objects is created through the semantic object abstractions in the semantic level 142.

The intelligent data assimilation system views data fields as the most basic elements of any data set retrievable by the intelligent data assimilation system. ***Prior to the ontological definition, these data fields generally include no semantic context or intelligence.*** Character or byte strings, numbers, decimals and dates can be identified; however, no meaningful definition is applied providing understanding of the data. For example, position 1 to 9 in a data field may include the value "561929975" but a typical data assimilation system does not understand whether this is a social security number, order number, or product number. ***Associating a semantic context to the data field by an intelligent data assimilation system results in an atomic object.***

Atomic objects have a semantic context associated with a value of a particular type. Thus, data fields can be translated into an atomic object. For example, the value might be a string "John Oct. 9, 2000 21:23.234 8A56FB" and include semantic information in the format: First Name, Date, and Result of Query ID.

Semantic objects are groupings of atomic objects, defined in the intelligent data assimilation system ontology, which have a particular business meaning. Intelligence is created by these higher-level groupings. For example, the following atomic objects: first name 147, middle initial 151, and last name 149 can be grouped to provide a "person name" semantic object 150. Additionally, the following atomic objects: address1 153, city 155, state 157, and zip code 159 are grouped to create a "person address" semantic object 152. Subsequently, a higher-level semantic object can be created by combining a "person name" with a "person address" to build a "person" semantic object 156. In a similar fashion, a credit file atomic object 148 is used to create a credit score semantic object 154. The credit score semantic object is then used to create a higher-level credit risk semantic object 158. ***Semantic objects are grouped in a data service level 144 of the intelligent data assimilation system ontology to create high-level data abstractions used by other sub-systems within in the intelligent data assimilation system.***

FIG. 5 is a diagram depicting an exemplary embodiment of an ontological relationship between semantic constructs and their associated LSOs in an ontology according to the present invention. The exemplary embodiment depicts a legal entity class 200 with two children classes, person 218 and corporation 202. Each of these classes includes a child class encapsulating the concept of

> where an instantiation of the legal entity class may live as defined for legal purposes. In the case of the person class, the appropriate class is the domicile class 220 and for the corporation the appropriate class is the corporate address class 204. The classes street address 212, city 210, Zip Code 208, and state 206 have multiple parents through multiple inheritance, namely the domicile and corporate address class.

Wachtel, col. 9, l. 7 through col. 10, l. 10 (emphasis supplied).

The cited portion of *Wachtel* teaches creation of an ontological model. Transformation from first data to second data is not yet being performed in the context of having also derived an ontology model. *Wachtel* does mention that data fields can be translated into "atomic objects," which is a step in *creating the ontological definition.* The ontological definition is later used to actually transform objects. Thus, *Wachtel* teaches exactly the opposite process as the invention of claim 1. *Wachtel* teaches using an ontological definition to perform a transformation, whereas the claimed "derived transformation" transforms data *without* using an ontological format.

Nevertheless, the examiner also cites the following portion of *Wachtel* as teaching this claimed feature:

> *The intelligent data assimilation system consolidates the results of all LSOs executed in a workflow and represents the resulting data as XML documents that can be delivered to a data client in that form. In one embodiment, the intelligent data assimilation system reformats the XML documents to create formatted outputs 124 using XSL stylesheets 125. These XSL stylesheets provide for the rendering of formatted data, including the simultaneous delivery of results in HTML 126, XML, Wireless Application Protocol (WAP) 128, Rich Text Format (RTF), Word (DOC), WordPerfect (WPS), or other formats 130.*
>
> An intelligent data assimilation system ontology facilitates the exchange of data between multiple computer systems independently of the individual system technologies, information architectures and application domain, and allows the creation of groupings of atomic data fields into intelligent building blocks termed semantic objects allowing the construction of services provided by the intelligent data assimilation system.
>
> An ontology includes a vocabulary of terms and a specification of what those terms mean. The intelligent data assimilation system ontology provides a set of well-founded constructs that are used to build meaningful higher-level knowledge. Basic terms in the intelligent data assimilation system ontology are selected such that basic foundational concepts and distinctions are defined and specified. The basic terms chosen form a complete set, whose relationship to one another is defined using formal techniques. These formally defined relationships provide the semantic basis for the terminology chosen. It is these relationships included in the intelligent data assimilation system ontology that enables the expression of domain-specific knowledge without including domain-specific terms.

> An intelligent data assimilation system ontology includes definitions of abstractions. For example, a person semantic is constructed by the underlying atomic or semantic items, first name, middle name, and last name. The definition is used to create a knowledge instance of the ontology, for example when at runtime the ontology is populated with specific semantic and atomic instances returned by the LSOs.

Wachtel, col. 7, ll. 4-14 (emphasis to show portions cited by the examiner).

The examiner's quotation is taken out of context. In the broader surrounding context, one of ordinary skill will immediately recognize that the *transformation used to transform XML formats using XLS stylesheets* ***is not a derived transformation that is derived using the ontology model.*** Instead, the transformed XML formats are used *in creating the ontological definition.* The ontological definition, in turn, *is used to transform the data.*

For these reasons, the teachings of *Wachtel* in this regard are inappropriately applied to claim 1. Claim 1 requires that the derived transformation be derived using an ontology model. However, claim 1 also requires that the derived transformation can transform the first data into the second data without transforming the data via an ontological format. *Wachtel* does not teach this claimed feature and instead only teaches deriving an ontological definition and then using that definition to perform the transformation *in an ontological format.*

Nevertheless, the examiner also cites the following portion of *Wachtel*:

> A workflow instance can call multiple LSOs either sequentially or in parallel. In this example, a first LSO 728 is called at step 722 and an invocation message in the form of an XML document 724 is sent to the LSO. The first LSO transmits data request messages 730 to a data provider 732 and receives requested data messages 734 from the data provider. The LSO responds by sending the requested data messages back to the workflow instance.
>
> The workflow instance determines if a complex search is being performed at step 738. If the search is a simple search and only the first LSO is to be called, then the workflow instance ends its search process at step 740 and transmits a status complete message 742 to the workflow state manager. If the workflow instance completed the search, a updated result XML document 764 is sent to the fulfillment line. If the workflow state manager determines that the workflow instance did not complete normally, it throws an exception 760 to the intelligent data assimilation system notifier 762.
>
> If a complex search is requested, such as using several LSOs to query separate databases or an LSO will be collecting data from a manual process, the workflow instance invokes a second LSO 744 and sends a second semantic context message 736 in the form of a XML document to a second LSO (not shown). Clients using services from an intelligent data assimilation system may request a fulfillment to be performed within the same session as the request, as in the case of real-time searches. To facilitate this, an intelligent data assimilation system has services that are configured to process the request synchronously. For

> searches that cannot be immediately fulfilled (as in the case of manual searches) an intelligent data assimilation system has services that are configured to process asynchronously. An intelligent data assimilation system, therefore, has the ability to synchronize a response message to the completion of workflow and include the resulting output in a response message. When an asynchronous service is requested, an intelligent data assimilation system responds to the data client independently of the completion of the workflow with the idea that the data client will "come back later" to request the results.

Wachtel, col. 14, l. 42 through col. 15, l. 13.

This portion of *Wachtel* is irrelevant to claim 1. The cited portions of *Wachtel* describe synchronous invocation of LSOs. LSOs are logical search objects that encapsulate knowledge and capabilities used to execute a search for data from a plurality of data providers 104. The intelligent data assimilation system is accessed by a data client via the communications network through the application server. See col. 5, ll. 27-32. *Wachtel* describes LSOs as follows:

> A LSO 106 is a reusable, configurable and self-contained software component capable of establishing connectivity to a particular data set provided by a data provider. Furthermore, a LSO includes logic for qualifying or filtering a data set and delivering the resulting information as a document written in extensible Markup Language (XML) or as an intelligent data object herein termed a semantic object 108. Additionally, a LSO reveals its capabilities and information to other software components for use by those software components.

Wachtel, col. 5, ll. 45-55.

However, LSOs are, themselves transformations *in an ontological format*. For example, *Wachtel* also provides that:

> *An ontology designer 122, is used to create the data mapping of data from a data provider into an intelligent data assimilation system ontology. It is through this process that the intelligent abstractions of data are performed to create semantic objects used by the LSOs maintained in the intelligent data assimilation system repository.*

Wachtel, col. 6, ll. 27-33 (emphasis supplied).

Because *Wachtel* teaches that LSOs use semantic objects, *in an ontological format,* to perform transformations of one form of data to another, *Wachtel* specifically teaches the opposite of claim 1. In stark contrast, claim 1 requires that the derived transformation can "transform the first data into the second data without transforming the first data via an ontological format." Because *Wachtel* does not teach this claimed feature, *Wachtel* does not anticipate claim 1.

Nevertheless, the examiner cites Figure 9 of *Wachtel* as teaching this claimed feature. Figure 9 is as follows:

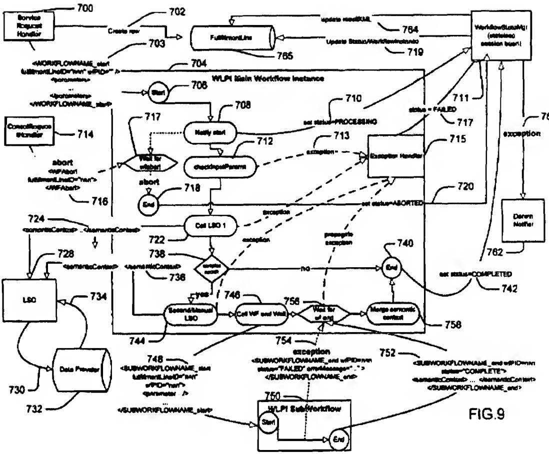

FIG.9

Figure 9 of *Wachtel* is a collaboration diagram depicting the interactions within an embodiment of a workflow process. The workflow process uses LSOs, as described above. Also as described above, LSOs are transformations that perform transformation via an ontological format. Therefore, Figure 9 teaches the opposite of claim 1. Accordingly, Figure 9 *Wachtel* does not teach all of the features of claim 1.

Nevertheless, the examiner cites Figure 4 of *Wachtel* as teaching this claimed feature. Figure 4 is as follows:

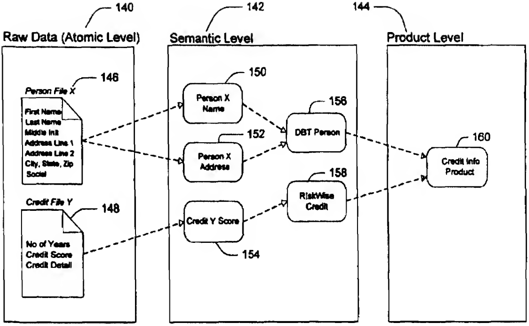

FIG. 4

Figure 4 of *Wachtel* is a diagram of the levels within one embodiment of an intelligent data assimilation system ontology. The system ontology is used by LSOs to accomplish transformation of data. Thus, as described above, LSOs are transformations that perform transformation via an ontological format. Therefore, Figure 4 teaches the opposite of claim 1. Accordingly, Figure 4 *Wachtel* does not teach all of the features of claim 1.

Applicants have proved that *Wachtel* does not teach the claimed feature of, "the derived transformation can transform the first data into the second data without transforming the first data via an ontological format." Instead, *Wachtel* teaches the opposite process – using an ontological format to accomplish a transformation. Therefore, *Wachtel* does not anticipate claim 1.

Independent claims 34 and 135 contain features similar to those presented in claim 1. Therefore, *Wachtel* does not anticipate claims 34 and 135. Claims 2-8, 20, 24, 26-33, 35-41, and 136 depend from one of the independent claims. Therefore, *Wachtel* does not anticipate these claims at least by virtue of their dependence from the independent claims. Accordingly, the rejection of claims 1-8, 20, 24, 26-41, and 135-136 has been overcome.

Furthermore, *Wachtel* does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. Instead, *Wachtel* teaches away from the claimed inventions because *Wachtel* teaches performing transformations via an ontological format instead of without transforming the first data via an ontological format, as claimed. Absent the Examiner pointing out some teaching or incentive to implement *Wachtel* and discarding queued connectionless datagrams as claimed,

one of ordinary skill in the art would not be led to modify *Wachtel* to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify *Wachtel* in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the applicants' disclosure as a template to make the necessary changes to reach the claimed invention.

IV. 35 U.S.C. §103, Asserted Obviousness

The Examiner rejected claims 9-19, 21-23, 25, and 42-69 as obvious over *Wachtel* in view of *Lindberg* et al., Method and System for Transferring Information Between a User Interface and a Database Over a Global Information Network, U.S. Patent 6,732,109 (May 4, 2004) (hereinafter "*Lindberg*"). This rejection is respectfully traversed.

A *prima facie* case of obviousness is established when the teachings of the prior art itself suggest the claimed subject matter to a person of ordinary skill in the art. *In re Bell*, 991 F.2d 781, 783, 26 U.S.P.Q.2d 1529, 1531 (Fed. Cir. 1993). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994).

As shown below, the proposed combination, when considered as a whole, does not teach or suggest all of the features of the independent claims. Therefore, the examiner has failed to state a *prima facie* obviousness rejection of claims 9-19, 21-23, 25, and 42-69 at least by virtue of their dependence on the independent claims.

The rejection rests on the false assumption that *Wachtel* teaches the bulk of the claimed features, except for those features characterized by the examiner as being taught by *Lindberg*. However, as shown above, *Wachtel* does not teach the claimed feature of, "the derived transformation can transform the first data into the second data without transforming the first data via an ontological format." Instead, *Wachtel* teaches the opposite process – using an ontological format to accomplish a transformation. Therefore, *Wachtel teaches against* the claims and does not suggest this claimed feature.

Additionally, *Lindberg* does not teach or suggest this claimed feature. The examiner does not assert otherwise. Instead, *Lindberg* teaches transferring information between databases on different computers, wherein the databases have four different processing layers. In fact, *Lindberg* is devoid of disclosure with respect to ontological transformations. Accordingly, *Lindberg* does not suggest this claimed feature.

Because neither *Wachtel* nor *Lindberg* teach or suggest the claimed feature of, "the derived transformation can transform the first data into the second data without transforming the first data via an ontological format," the proposed combination of these references when considered as a whole does not

teach or suggest all of the features of the dependent claims. Accordingly, under the standards of *In re Lowry*, the examiner has failed to state a *prima facie* obviousness rejection against the dependent claims.

Additionally, no teaching, suggestion, or motivation exists to combine *Wachtel* and *Lindberg* because the references address wholly different problems. It is necessary to consider the reality of the circumstances--in other words, common sense--in deciding in which fields a person of ordinary skill would reasonably be expected to look for a solution to the problem facing the inventor. *In re Oetiker*, 977 F.2d 1443 (Fed. Cir. 1992); *In re Wood*, 599 F.2d 1032, 1036, 202 U.S.P.Q. 171, 174 (CCPA 1979). In the case at hand, the cited references address distinct problems. For this reason, no common sense reason exists to establish that one of ordinary skill would reasonably be expected to look for a solution to the problem facing the inventor. Accordingly, no teaching, suggestion, or motivation exists to combine the references and no *prima facie* obviousness rejection can be made against the dependent claims using *Wachtel* and *Lindberg*.

For example, *Wachtel* is directed to solving the problem of scalability and reusability of a Web server architecture. *Wachtel* provides that:

> Previous development of Web servers focused on producing the most efficient use of computing resources to produce finer and finer vertical slices through a Web site. For example, a servlet once invoked includes all of the necessary logic necessary to acquire data according to a client request, format the acquired data into a document, and report back to the hosting system any statistics about the client/servlet interaction. Finer and finer vertical slices through a Web site leads to an unscalable Web server architecture because all the logic for acquiring data, generating a document, and producing side effects is included within a monolithic software object thus defeating the ability to reuse or integrate a particular software object in another system or software object.
>
> Therefore, a need exists for a scalable Web server architecture that features a high degree of reusability of a Web server's components. The present invention meets such a need.

Wachtel, col. 1, l. 59 through col. 2, l. 9.

On the other hand, *Lindberg* is directed to the problem of making abstract database architectures understandable and usable. For example, *Lindberg* provides as follows:

> In the prior art systems, there is no, or poor separation of concern between the structure of information and the structure of its use. In the project example discussed above, although the information structures (Car and Project) were almost identical, the implementation included project semantics in the information model. In other words, each implementation included structures that were specific to each particular application. The best solution would have been an abstract information model. Unfortunately, when using an "abstract model," the developer writing the business logic must use the structures of the model which, because they are "abstract", necessarily have abstract and difficult names. Therefore, the developer cannot express the logic in well-known terms.

> Moreover, because the abstract structures are generic to multiple uses, the developer may need to use several layers of "mental indirection" to obtain the desired result. Therefore, while an abstract information model is very flexible, its abstract nature makes the resulting logic difficult for users and subsequent developers to understand. Separation is needed between the need for an information model capable of holding all of the information in a flexible (adaptable and changeable) way, and the need for the users of the information to understand and work with the information.

Lindberg, col. 2, ll. 22-52.

Based on the plain disclosures of the references themselves, the references address completely distinct problems that are unrelated to each other. The problem of scalability and reusability of a Web server architecture is completely distinct from the problem of making abstract database architectures understandable and usable. Because the references address completely distinct problems, one of ordinary skill would have no reason to combine or otherwise modify the references to achieve the claimed invention. Thus, one of ordinary skill in the art would not combine these references to achieve the invention of the dependent claims, because no teaching, suggestion, or motivation exists to combine the references in the manner suggested by the examiner. Accordingly, the examiner has failed to state a *prima facie* obviousness rejection against the dependent claims.

V. **Conclusion**

The subject application is patentable over the cited references and should now be in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: November 30, 2006

Respectfully submitted,

/Theodore D. Fay III/

Theodore D. Fay III
Reg. No. 48,504
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants